Zoea

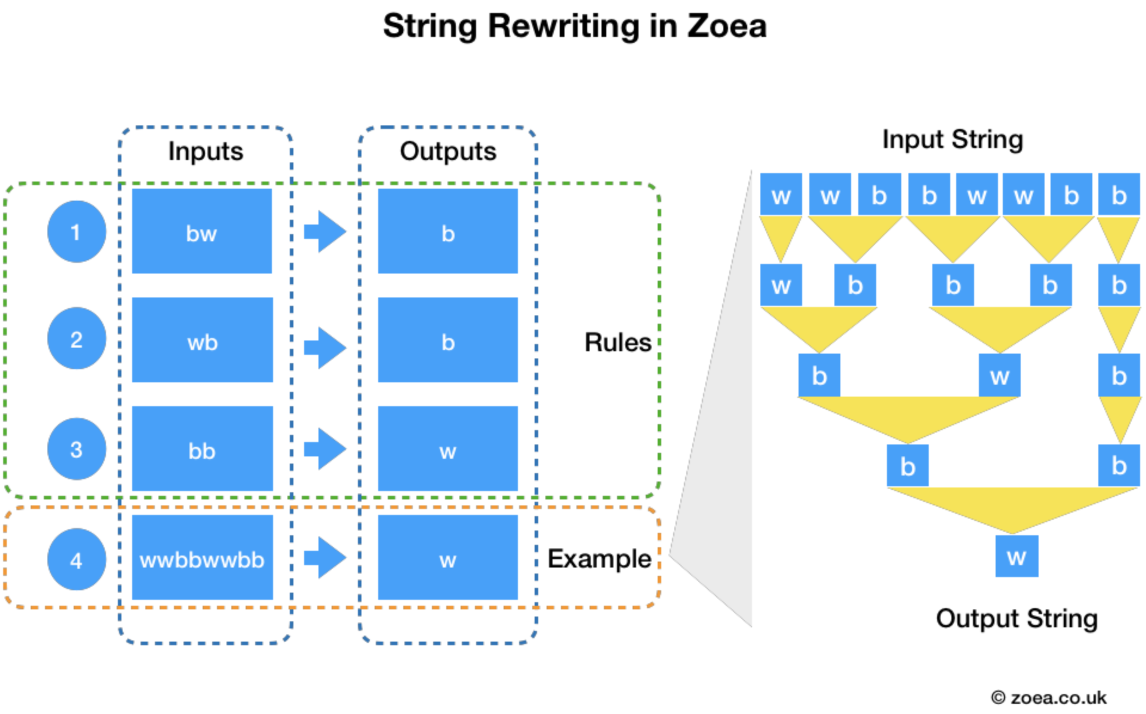
Research Note 14

String Rewriting in Zoea

Edward McDaid & Sarah McDaid

07 Aug 2021

String rewriting systems are a type of program that converts text according to a set of rules. They have a wide range of practical uses but also serve as an important theoretical model for computer science. So how would you code a string rewriting system in Zoea?



String rewriting is a simple concept. You are given a piece of text and a set of rules. Each rule simply states that if you find a given substring anywhere in the text then replace it with some different text. Rewriting is simply the process of applying each of the rules in turn to the text until you can't apply any more rules. Then you stop and what you are left with is the output.

This is similar in concept to search and replace in a word processor except that we can have any number of pairs of from/to values. Another difference is that each time a substring is replaced the program starts again from the beginning of the text and also from the first rule. This means that the order of the rules is significant.

String rewriting has some obvious applications in areas such as text processing, linguistics and network protocols. It can also be used to create systems that manipulate algebra, implement simple AI reasoning and even create computer art. One interesting property of string rewriting systems is that they are Turing complete. In other words rewriting is effectively a form of programming language that can - at least in theory - be used to build any program that you could write for a universal Turing machine. As a result rewriting has been used as the basis for some interesting programming languages.

To demonstrate how to implement a string rewriting system in Zoea we first need an example. We will use a simple game with a single row of black and white pebbles. There are three rules. 1) Replace any adjacent pair that is black-white with a single black pebble; 2) Replace white-black with black; 3) Replace black-black with white. We apply these rules repeatedly for as long as we can. Given a random sequence of black and white pebbles of any length the object of the game is to determine the end state.

We can easily represent the state of such a game as a string with characters b and w standing for black and white respectively. In Zoea each transformation rule can simply be expressed as a single test case by specifying the input and output values. For example input 'bw' gives output 'b'. If we had an instance of the game where this was the input configuration then the single rule would give the correct output. However the three rules on their own provide no indication to Zoea that this is a rewriting problem. We do that by providing one or more examples that show a complete application of the rules to transform an input into an output. Case 4 in our example has input "wwbbwwbb" and output "w" although any example that uses all of the rules would also suffice. The complete Zoea program is as follows.

```
program: rewrite
case: 1 input: bw output: b # rule 1
case: 2 input: wb output: b # rule 2
```

```
case: 3 input: bb output: w # rule 3
case: 4 input: wwbbwwbb output: w # example
```

Each of the three rules are perfectly valid cases in their own limited scope. They describe how a single transformation proceeds using a single rule and an input in which just that rule can be applied once. The examples are also valid cases but show what the result would be if all of the rules are applied repeatedly. From the Zoea developer perspective the examples can be regarded as unit tests for the complete rewriting rule base. Zoea on the other hand recognises the examples as cases where all the rules are being applied and where recursion is also used.

We can have any number of rules and examples but we need to have at least one of each. The rules and examples do not need to be in any particular order as Zoea is able to determine which is which. String rewriting can also be used as part of a more complex program to produce an intermediate value or in a subsidiary test case. Using the test cases in a Zoea program as a set of rules is a powerful technique that can be used to describe complex requirements in a way that is both simple and accessible.

Learn more at **zoea.co.uk**